




# Fake News Detection

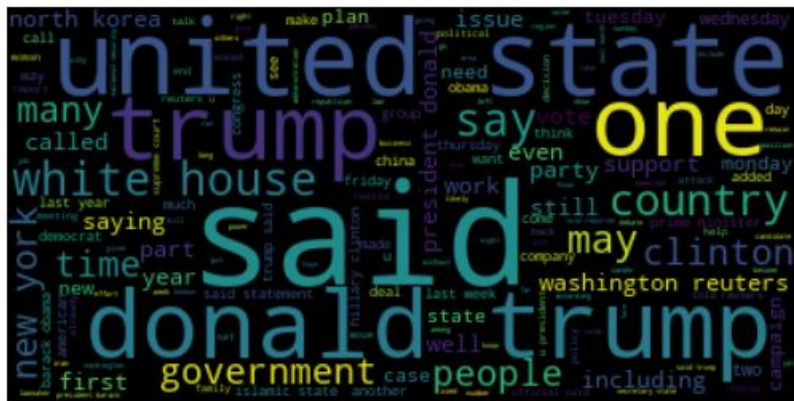
Karolina Mączka, Tymoteusz Urban



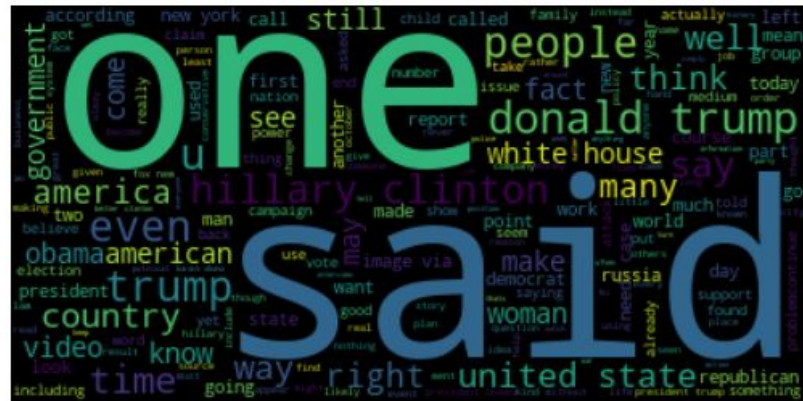
# DATA EXPLORATION

Data: dataset with three columns: title, text, label

Goal: classify texts as true or fake



## Most common words from true texts



## Most common words from fake texts

# PREPROCESSING

- Encoding labels
- Combining columns
- Cleaning texts (links, punctuations)
- Dropping non english languages
- Removing stopwords
- Lemmatizing words
- Applying CountVectorizer
- Using TfidfTransformer

```
df3[df3['lang'] != 'en']['label'].value_counts()
```

```
0    254  
Name: label, dtype: int64
```

All non english texts were fake

# FEATURES EXTRACTION

We can see that 'reuters' feature is very important but it shouldn't be for a universal model as it is the source of the news.

It is highly connected with true texts because of how the dataset was created.

Deleting it is gonna make our model worse on this data, but it will be more useful for other datasets.

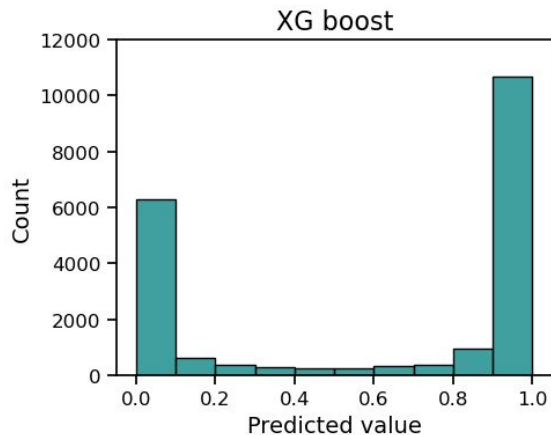
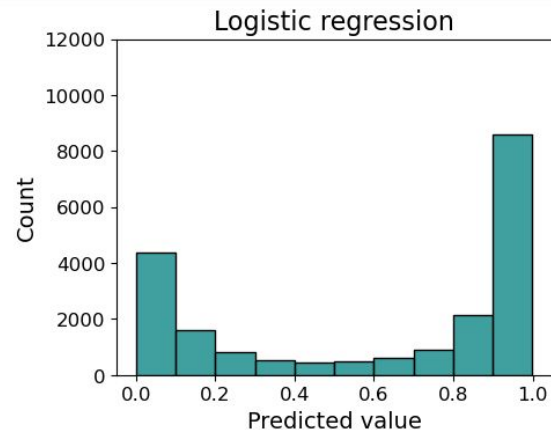
Top 10 features with highest importance

	Importance
reuters	0.168121
reenter	0.012295
thats	0.009026
contest	0.008996
startup	0.008580
gop	0.007781
via	0.007457
sen	0.006414
qualifying	0.006268
killing	0.006141

# CHOOSING CLASSIFIER

Results on test set	precision	recall	f1 score	auc score	log loss
Classifier					
Logistic regression	0.942938	0.964566	0.954315	0.983883	0.179598
Multinomial Naive Bayes	0.922952	0.961841	0.895908	0.936989	0.351124
SVM	0.931106	0.959836	0.932114	0.902366	NaN
SVM with bigrams	0.940989	0.968815	0.916572	0.871418	NaN
SVM with huber	0.934757	0.960077	0.945635	0.925197	NaN
XG boost	0.945293	0.965528	0.959605	0.988827	0.131223

We will work with XGboost as it does best in all metrics



# OPTIMIZING MODEL

	precision	recall	f1 score	auc score	log loss
Classifier					
XG boost	0.945293	0.965528	0.959605	0.988827	0.131223
XGB no tfidf	0.947328	0.966731	0.961528	0.988959	0.131162
XGB with bigrams	0.945381	0.965208	0.962584	0.989553	0.125730

Practically there is no difference, so we will use Tfidf as it makes model more sensible and explainable. We also won't include bigrams as they make data dozen times bigger, but hardly affect the results.

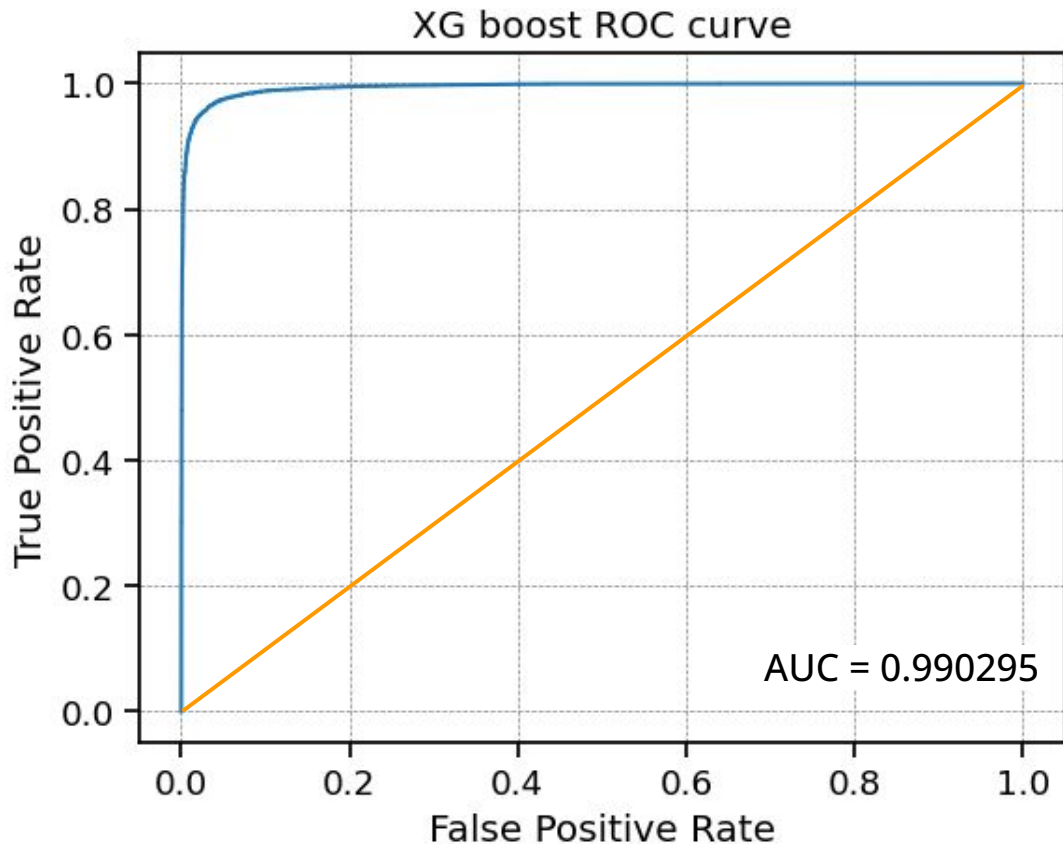
	precision	recall	f1 score	auc score	log loss
Classifier					
XG boost	0.945293	0.965528	0.959605	0.988827	0.131223
XGB boost final	0.949747	0.968174	0.963770	0.990295	0.124402

Thanks to randomized search on hyper parameters with cross validation we managed to slightly improve our model

# FINAL MODEL

We can see that our model is really good at predicting fake news.

Validation confirms it as there is  $AUC=0.988725$  on validation set.



# VALIDATION - main concerns

- Handling nans
  - We combine columns instead of deleting rows with nans
- Deleting too much data
  - Went from deleting 10% of rows because of badly done language detection to only 0.5%
- Hyper parametrization
  - Switched from manual parameter search to random search with CV